

Jack
FYI
JC

Probing the news

Data communications

John
Couch
FYI
ST

Can software do encryption job?

In the face of standards agency's recognition of hardware only
and objections of chip makers, versions are appearing

by Deborah Williams, McGraw-Hill Publications Co., and Harvey J. Hindin, Communications & Microwave Editor

When the National Bureau of Standards published its data encryption standard (DES) three years ago, it stated that only hardware implementations would be certified. Although there was nothing in the algorithm precluding it, software was considered too slow, difficult to validate, and subject to unauthorized modification. So hardware manufacturers took advantage of the opportunity to develop special DES chips.

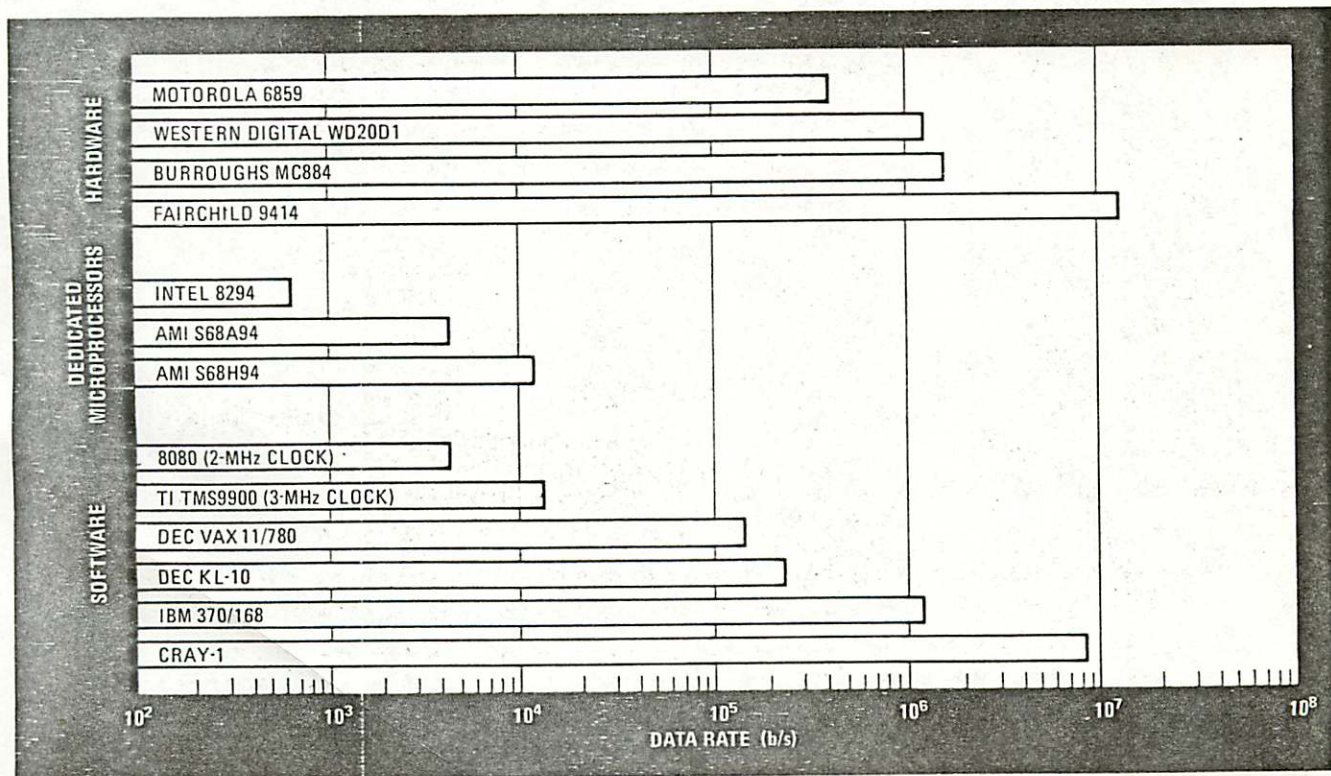
But undaunted by the lack of NBS blessing, software experts continued to work on their versions and the results are coming in. For example, Richard Gumpertz, an associate in

the computer science department at Carnegie-Mellon University in Pittsburgh, has implemented a DES program for a variety of computers that gives results as good as or better than those of existing hardware.

Not everyone is happy with this development. First, there is an ongoing controversy over why the NBS did not certify software in the first place; also, hardware manufacturers question software's security and decry its impact on their market. On the bright side, industry experts agree that fast software will expand the sluggish encryption market itself [*Electronics*, June 5, p. 96].

Not against software per se, the NBS "uses software for DES testing," says Dennis Branstad, director of NBS computer security there. He adds that "it's not impossible that the bureau would consider a DES for software."

Why the delay? The NBS originally estimated that the software implementation of a standard 64-bit data block would take 30 to 200 microseconds. That is the equivalent of a data rate of 0.3 to 2.0 kilobits per second, which is not at all suitable for high-speed communications or computer links. But Gumpertz determined that one of the reasons a programmed



Fast approaches. Gumpertz's program yields throughputs in the popular block-cipher mode that are on a par with hardware speeds (compare bottom six bars with top seven). The time to get data to and from the computer is not included in the calculation, nor are computer interrupts.

implementation is slow is that the typical computer is not equipped with the right data paths for the necessary DES algorithm manipulations.

Making fast software harder to develop are the initial and final permutation steps in the algorithm. These two mathematical manipulations, difficult to implement in software, greatly increase a program's execution time. Gumpertz suggests that they were included in the DES both to ease some hardware designs and to discourage software.

The algorithm's 16 internal expansion cycles, each of which requires eight arithmetic manipulations, present another obstacle. For Gumpertz this was the principal bottleneck to a fast program and took up more than half the execution time. "The instruction sets of typical computers do not offer much assistance in implementing the expansion operations," he explains.

Gumpertz's program finally overcame all these obstacles without needing much computer space. Only 150 bytes are required for the data, and 2.5 kilobytes are needed for the program. "The program does not take up much space because most of the time is spent doing the same thing over and over," he says. "Furthermore, few temporary locations are needed for intermediate results," he concludes.

Same difference. For Steven Kent, too, software is the way to go. Asked to compare his approach with Gumpertz's, the research assistant at the Massachusetts Institute of Technology in Cambridge, Mass., says that his group's program does not combine operations in the same way nor does it have the problem with the expansion cycles.

"Our timing is fairly compatible with Gumpertz's but our implementation doesn't have the same combination of operations. I assure you that the expansion steps don't have to be a limitation to a fast DES. Gumpertz is doing it just one way of the many possible," he says. Kent's group sold an IBM System/370 program to a mainframe manufacturer for \$10,000, but he says that similar versions could be sold for less.

In contrast to Gumpertz and Kent, Herbert Bright, president of

Computation Planning Inc. in Bethesda, Md., has been marketing software-encryption systems since 1975. His \$4,700 Desqik package does the algorithm at 38 kilobytes per second on machines in the Amdahl 470V/6 class. He says business in the first four months of 1980 was better than for the last four and a half years, although he will not give figures.

Of course, some of Gumpertz's assumptions are challenged by the hardware people. Says Ken Cohen, security product line manager at Western Digital Corp. of Newport Beach, Calif., "I think Gumpertz has demonstrated that it is possible to write an efficient DES program, but I question how fast some of the subroutines—particularly in the Cray-1—would be in an interrupt environment. In the real world, that machine has to be doing things such as collecting data from a communications line or a disk, decrypting it, and passing it off to an applications program. What happens to the speed of the algorithm in that environment, what happens when encryption is fighting for priority in the stack of jobs to be run and the processor is handling many input/output interrupts?"

Occasional. Cohen also questions the enormous cost of running a machine when inexpensive hardware is available. Gumpertz acknowledges that it may not be fair to compare the \$10 million Cray to a chip. But he suggests that a general-purpose processor would be more economical if encryption is not often used.

The occasional-use problem is only one of several reasons to consider software in the first place. Perhaps more important, according to Gumpertz, the appropriate interfacing circuitry may not be available.

Gumpertz adds that once software encryption is installed, it can be used on data other than that being transmitted or received over one particular channel. "This flexibility can be very useful. Consider, for example, data on a magnetic tape. Even if there is already DES hardware on site, it probably has been wired into the communications system and so cannot be used for this purpose. On the other hand, a subroutine can trivially be used to massage data before writing it on tape," he says. □

Im agi na tion

is just one of our advantages. Others are reliability, versatility and combat-proven use.

We offer you custom-designed electronic equipment and systems for military microwave communications (command, control, data) and power.

Try us for communications systems

- airborne and other military power supplies and converters •
- pulse radar altimeters •
- proximity fuzes for artillery and mortars.

Tell us what you need.

telkoor LTD
electronics industries

Marketing Manager
P.O.B. 76, Petah Tikva, Israel
Tel. 03-903661-3,
Telex: 341993 MAAC IL.

A subsidiary of Koor Electric & Electronics Ltd.

The Mathematics of Public-Key Cryptography

The search for privacy in an age of electronic communications has given rise to new methods of encryption. These methods are more practical than older ones and are mathematically more interesting

by Martin E. Hellman

The electronic communications systems that are proliferating throughout modern society offer speed, accuracy and ever diminishing cost. They also present serious problems of security. As the ordinary transactions conducted in person, on the telephone or by written correspondence have come increasingly to be conducted by new kinds of electronic systems the susceptibility of organizations and individuals to eavesdropping and forgery has grown dramatically. One way to prevent tampering with the new electronic systems and to protect the vast quantities of private information such as the credit rec-

ords and medical histories now stored in computer data banks is to resort to cryptosystems: methods for encrypting, or transforming, information so that it is unintelligible and therefore useless to those who are not meant to have access to it.

Encryption is a special form of computation, and almost all modern cryptosystems depend on difficulty of computation for their security; they effect transformations of data so complicated that it is beyond the economic means of an eavesdropper to reverse the process. (Accounts of intelligence operations during World War II reveal that as re-

cently as 35 years ago systems offering this type of security were not widely available. Since then the cost of computation has dropped by a factor of about a million, so that the equipment necessary for secure encryption is now reasonably priced.) Given unlimited computing power (an unrealistic assumption) such computationally secure systems could be broken, but in practice they appear to be unbreakable.

At present mathematicians lack the tools for proving systems to be computationally secure, and the history of cryptography demonstrates all too well that supposedly unbreakable systems often have hidden flaws. It is hoped that discoveries in complexity theory, a branch of mathematics that studies the difficulty (or cost) of computation, will eventually provide the tools needed to establish provably secure cryptosystems: computationally secure systems that can be guaranteed to be free of hidden flaws. In the meantime a group of mathematical problems characterized by a certain kind of computational intractability are serving as the basis of a new class of encryption procedures that are in many ways superior to current techniques. The proposed new systems, which were first put forward by Ralph Merkle, Whitfield Diffie and me at Stanford University, are termed public-key cryptosystems. To understand the significance of the term it is necessary to consider briefly how methods of encryption have developed historically.

Any cryptographic technique, such as the substitution and transposition of symbols, that operates on a message without regard to its linguistic structure is called a cipher and is said to generate a ciphertext. (Codes, which I shall not discuss here, operate on larger linguistic units such as words or phrases.) More precisely, the basis of any cipher is an invertible function: an operation (performed by the sender of the message) that converts a plaintext, or unenci-

A = 00000	B = 00001	C = 00010	D = 00011	E = 00100
F = 00101	G = 00110	H = 00111	I = 01000	J = 01001
K = 01010	L = 01011	M = 01100	N = 01101	O = 01110
P = 01111	Q = 10000	R = 10001	S = 10010	T = 10011
U = 10100	V = 10101	W = 10110	X = 10111	Y = 11000
Z = 11001	= 11010	= 11011	= 11100	? = 11101
∴ = 11110	∴ = 11111			

a = 00	b = 01	c = 02	d = 03	e = 04	f = 05	g = 06
h = 07	i = 08	j = 09	k = 10	l = 11	m = 12	n = 13
o = 14	p = 15	q = 16	r = 17	s = 18	t = 19	u = 20
v = 21	w = 22	x = 23	y = 24	z = 25	A = 26	B = 27
C = 28	D = 29	E = 30	F = 31	G = 32	H = 33	I = 34
J = 35	K = 36	L = 37	M = 38	N = 39	O = 40	P = 41
Q = 42	R = 43	S = 44	T = 45	U = 46	V = 47	W = 48
X = 49	Y = 50	Z = 51	0 = 52	1 = 53	2 = 54	3 = 55
4 = 56	5 = 57	6 = 58	7 = 59	8 = 60	9 = 61	= 62
= 63	= 64	∴ = 65	? = 66	...		

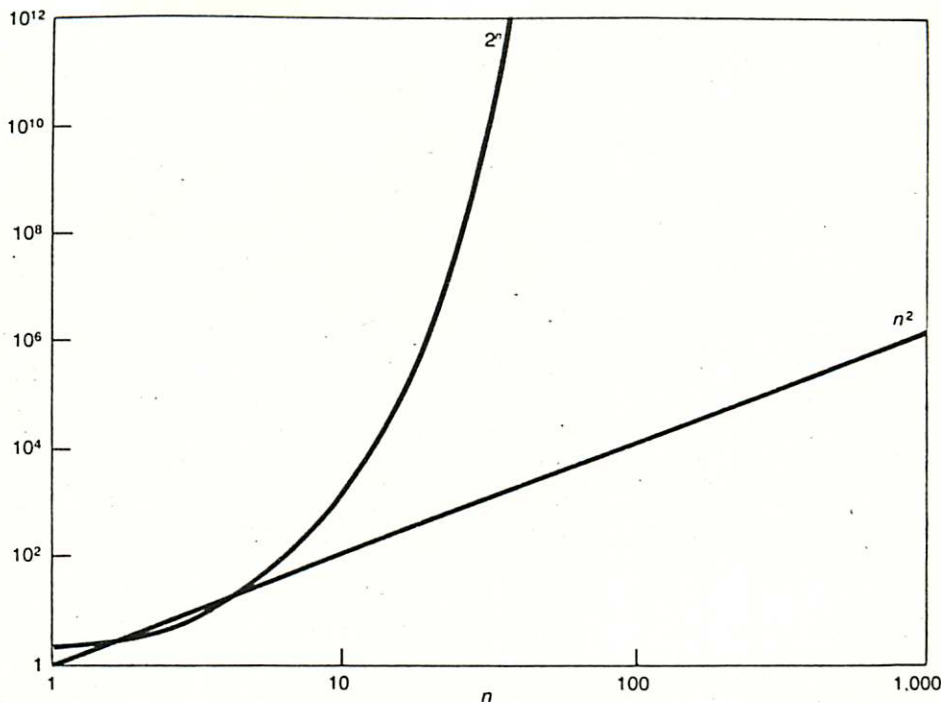
CRYPTOGRAPHIC SYSTEM is a mathematical system for encrypting, or transforming, information so that it is unintelligible and therefore useless to those who are not meant to have access to it. The encryption process generally begins with the conversion of the plaintext, or unencrypted message, into a string of numbers by means of a digital "alphabet" such as one of those shown here. In some cryptosystems it is more convenient to work with binary numbers, and so in the rather simple alphabet shown at the top five bits (binary digits) have been allocated to represent each letter, number or punctuation mark in the plaintext. Each bit can take two values (0 or 1), making a total of 2^5 , or 32, characters in this alphabet. In other cryptosystems it is simpler to think in terms not of a binary (base-2) number system but a decimal (base-10) one. In alphabet shown at bottom two decimal digits have been allocated for each plaintext symbol, providing total of 10^2 , or 100, characters. (Some of these may not be needed.)

C and the a_i 's. The receiver must solve the same knapsack problem, but to simplify the task he has additional information: his secret trapdoor parameters and deciphering key.

These steps should be made clear by a simple example [see illustration on page 155]. Consider a plaintext message in which the first word is HOW. In binary form the message begins 0011101-1101011011010. (This binary string, in which the last five-bit block represents the space between HOW and the next word in the message, is generated by the five-bit binary alphabet described above.) Now assume that the intended receiver's public enciphering key is $a = (2,292, 1,089, 211, 1,625, 1,283, 599, 759, 315, 2,597, 2,463)$, or $a_1 = 2,292, a_2 = 1,089$ and so on. Here n equals 10, and the first block of information, which consists of the first n bits in the binary plaintext, is $x = (0, 0, 1, 1, 1, 0, 1, 1, 1, 0)$. It is enciphered, then, as $C = a_1x_1 + \dots + a_nx_n$, or $C = (2,292 \times 0) + (1,089 \times 0) + (211 \times 1) + (1,625 \times 1) + (1,283 \times 1) + (599 \times 0) + (759 \times 1) + (315 \times 1) + (2,597 \times 1) + (2,463 \times 0)$. Therefore C equals 6,790, and to decipher the message it is necessary to determine which of the a_i 's add up to 6,790. (If a_i is included in the sum, x_i is 1 and vice versa.)

None of the known methods for solving the knapsack problem is substantially less time-consuming than conducting an exhaustive search, that is, adding up all the 2^n possible subsets of the a_i 's to see which subset yields C . In the example given above, where the number of elements n is equal to 10, this might be considered a workable approach. Someone intercepting C could try all the 2^{10} , or 1,024, possible combinations of the publicly listed a_i 's and thereby recover the vector x . In this instance the number of elements in a is too small to provide real secrecy. The knapsack problem is an NP one, however, and therefore the computational difficulty of all known solution methods rapidly "blows up." When the number of elements n is, say, 1,000, the number of possible subsets $2^{1,000}$ is greater than the number of atoms in the known universe. Deciphering by checking $2^{1,000}$ different subsets is quite impossible, and so C effectively shields the secret information x . On the other hand, enciphering 1,000 bits of information in this system is quite efficient, requiring no more than 1,000 additions.

So far I have described what appears to be a one-way function: apparently no one, including the receiver, will be able to recover x . If the elements of vector a are chosen at random, this is exactly the type of system that results. Even in this simple example, however, a trapdoor has been built in. The vector a has been structured in such a way that with a small amount of additional information



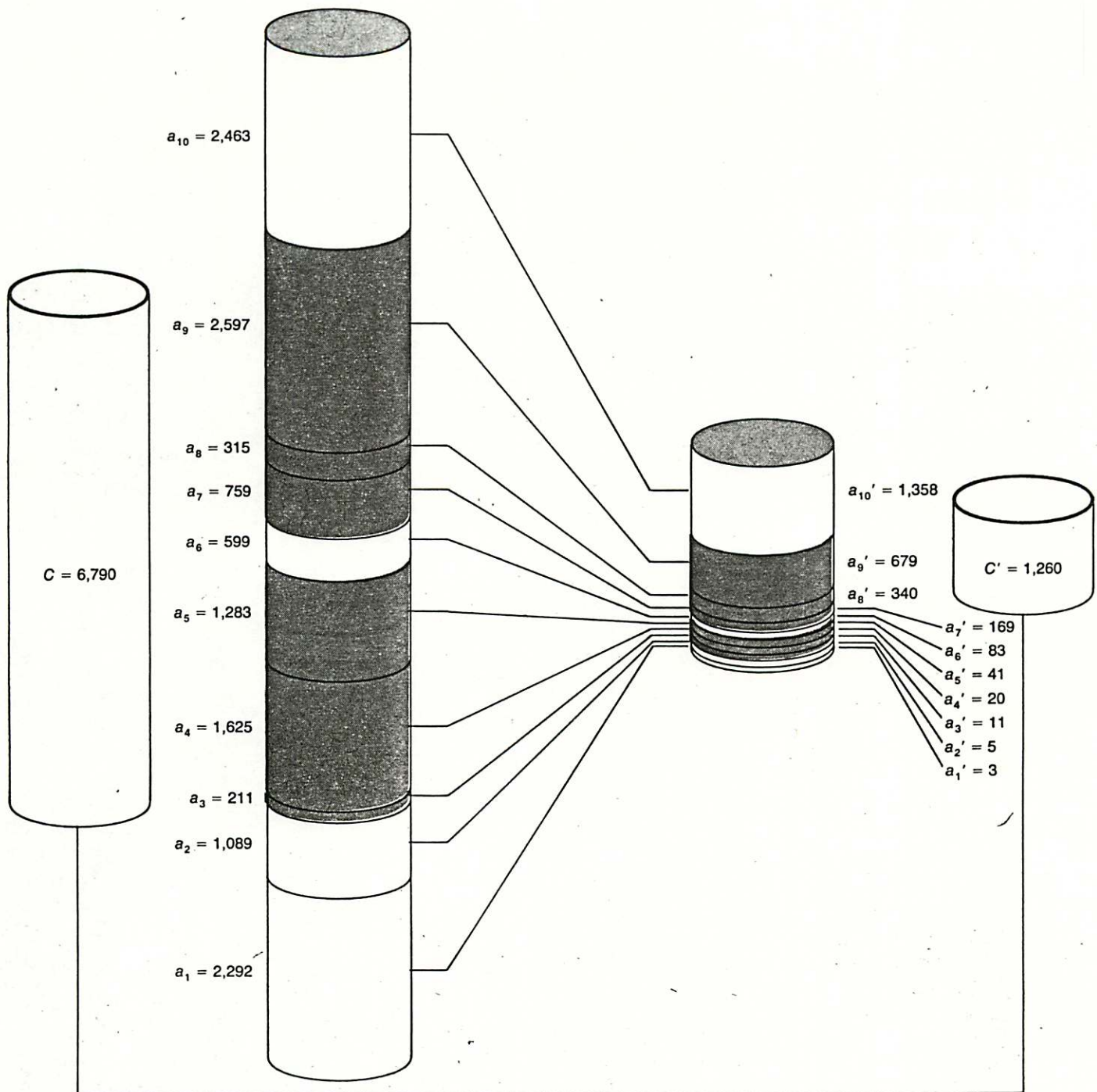
PROBLEMS IN THE CLASS NP (which stands for nondeterministic, polynomial time) are characterized by the fact that although it is easy to check a nondeterministic, or guessed, solution, it is hard to find a correct solution: As the size n of an NP problem increases, the number of computational steps and hence the time required to check a solution increase in proportion to a polynomial function of n such as n^2 (black curve), but all known methods of finding a solution increase in proportion to a more rapidly growing function of n , typically an exponential one such as 2^n (colored curve). Exponential functions increase far more rapidly, and when n is sufficiently large, NP problems become computationally infeasible. Hence they lend themselves readily to the design of one-way functions: easily computed functions whose inverses are infeasible to compute. In some cases such problems can be developed into trapdoor one-way functions: easily computed functions whose inverses are infeasible to compute unless certain facts employed in design of functions are known. Trapdoor one-way functions serve as basis of public-key cryptosystems: public key specifies easily computed function, which is infeasible to invert unless one knows secret key; that key specifies easily computed inverse function.

x can be derived from C much more rapidly than by an exhaustive search. As I have noted, not all NP problems lend themselves to the insertion of such a trapdoor. Here the trapdoor can be devised because there are certain vectors for which the knapsack problem is not difficult to solve. The receiver takes one of those special vectors a' and disguises it, publishing the resulting ordinary-looking vector a in the public file of enciphering keys. The trapdoor information enables him to move back and forth between a difficult knapsack problem involving a and the easy but equivalent knapsack problem involving a' .

To be more precise, in generating his public vector a the receiver begins by choosing a vector $a' = (a'_1, \dots, a'_n)$ in which each element a'_i is larger than the sum of the preceding elements $a'_1 + a'_2 + \dots + a'_{i-1}$. For example, if a' equals (3, 5, 11, 20, 41, 83, 169, 340, 679, 1,358), then a'_2 , which equals 5, is greater than a'_1 , which equals 3; a'_3 , which equals 11, is greater than $a'_1 + a'_2$, which equals 3 + 5, or 8, and so on. Now, consider a ciphertext $C' = 1,260$ that was generated with this special vector a' . In other words, C' equals $a' \cdot x'$ for some binary vector

$x' = (x'_1, \dots, x'_n)$, that is, 1,260 equals $3x'_1 + 5x'_2 + 11x'_3 + 20x'_4 + 41x'_5 + 83x'_6 + 169x'_7 + 340x'_8 + 679x'_9 + 1,358x'_{10}$.

Once again the problem of decipherment is equivalent to solving a knapsack problem, but in this instance because of the special property of the vector a' the solution x' is easily determined. To begin with, a'_{10} , which equals 1,358, is greater than C' , which equals 1,260, and so obviously cannot be part of the subset sum, that is, the rod is too long to fit into the knapsack. Hence x'_{10} must be 0. The next-largest element in the vector is a'_9 , or 679, which is less than C' , or 1,260. As the special property of a' dictates, the sum of the eight remaining elements of a' must be less than 679, and so those elements alone cannot "fill" the knapsack of length 1,260. Therefore 679 must be part of the sum, and x'_9 must be 1. Since x'_9 equals 1 and x'_{10} equals 0, the equation $C' = a' \cdot x'$ can now be rewritten as $1,260 = 3x'_1 + 5x'_2 + 11x'_3 + 20x'_4 + 41x'_5 + 83x'_6 + 169x'_7 + 340x'_8 + 679 + 0$. Subtracting 679 from both sides of the equation reduces the problem to determining which of the elements a'_1, \dots, a'_8 add up to $1,260 - 679$, or 581 (the length of the still empty part of the knapsack). Since a'_8



KNAPSACK PROBLEM is an NP problem from which a trapdoor one-way function can be derived. The cylinder and set of rods shown at the left illustrate the classic knapsack problem: Given a knapsack, or cylinder, of length C and a set of n rods all of the same diameter as the knapsack but of lengths a_1, a_2, \dots, a_n , find a subset of rods that fills the knapsack completely. This problem is in the class NP because the best method known for solving it is not much more efficient than trying all 2^n possible subset sums to see which one equals C , and yet a guessed solution can be checked with no more than n additions. Even in the small 10-rod examples shown here, finding a solution (color) by this method requires the testing of 2^{10} , or 1,024, different subsets, and when n is, say, 100, the task becomes impossible. An ordered set of numbers such as $a = (a_1, \dots, a_n)$ or $x = (x_1, \dots, x_n)$ is called a vector, and the "dot" product of two vectors $a \cdot x$ is defined as the sum $a_1x_1 + \dots + a_nx_n$. Given a fixed vector a , a function of the variable vector x can be defined as the dot product of x with the vector a , that is, $f(x) = a \cdot x$. If the elements x_1, \dots, x_n of x are all equal to 0 or 1, then inverting this function, or determining which value of x gives a particular sum $C = a \cdot x$, is equivalent to solving the knapsack problem for C and the given values of a_1, \dots, a_n . The function is one-way be-

cause the knapsack problem is in the class NP. Moreover, a trapdoor can be built into the function, because for certain vectors, or sets of rods, $a' = (a'_1, \dots, a'_n)$ the knapsack problem is easy to solve. In these sets, such as the one shown in the problem at the right, each element is greater than the sum of the preceding elements. To determine which subset fills the knapsack begin with the last, or largest, element a'_n . In this case a'_{10} equals 1,358, which is greater than 1,260, the length of the cylinder C' . Hence a'_{10} is not in the subset (that is, in the sum $C' = a'_1x_1 + \dots + a'_{10}x_{10}$, x_{10} equals 0). But a'_9 , which equals 679, is smaller than 1,260, and since the remaining elements in the set add up to a number even smaller than a'_9 , it must be in the subset (that is, x_9 equals 1). The problem is now reduced to filling the remainder of the cylinder, whose length is $C' - a'_9$, or 581, with a subset of the remaining rods a'_1, \dots, a'_8 , and so on. Continuing in the same way, the problem can be solved (or the function based on it can be inverted) with no more than 10 comparisons and 10 subtractions. As colored lines indicate, there is a way to move back and forth between the easy and hard knapsack problems. Parameters for effecting that transformation are secret trapdoor information for trapdoor one-way function based on knapsack problem (see illustration on page 155).

equals 340, which is less than 581, it is included in the sum. Thus x_8 is 1. Continuing in this manner, it can be determined that the x' is the original message block $x = (0, 0, 1, 1, 1, 0, 1, 1, 1, 0)$.

Constructing an easy knapsack vector such as a' is not difficult, but how does the receiver get from a' to a and back again? To accomplish that feat he chooses two large random numbers w and m and generates the vector a according to the equation $a_i = a'_i w \text{ modulo } m$, for each i from 1 to n . The expression "modulo m " indicates that a_i should be taken to be the remainder left when $a'_i w$ is divided by m . For example, if w equals 764 and m equals 2,731, consider the element a'_4 of vector a' . Since a'_4 equals 20, $a'_4 w$ equals 15,280. Dividing 2,731 into 15,280 gives 5 with a remainder of 1,625, so that a_4 , or $a'_4 w \text{ modulo } m$, equals 1,625.

Modular arithmetic plays a large part in public-key cryptosystems, because it turns smooth, or continuous and continually increasing or decreasing, functions into discontinuous ones, introducing a large factor of confusion into the calculation of their inverses: the values of x that correspond to particular values of $f(x)$. Consider the simple function $f(x) = 4x$. As x increases, the value of $f(x)$ increases in a very orderly way; for example, $f(3)$ is 12, $f(4)$ is 16, $f(5)$ is 20, $f(6)$ is 24 and so on. As a result if one is given a specific number $y = f(x)$, it is not difficult to determine x by a process of guesswork and elimination without ever actually solving the equation $y = 4x$. In other words, if a function is smooth, then no matter how hard solving explicitly for x is, it may still be possible to determine the value of x for a particular $f(x)$ through trial and error. For example, if $f(x)$ equals 20, one might guess that x equals 3. Then $f(x)$ would equal 12, which is too small, so that the correct value of x must be greater than 3. If, however, $x = 6$ were tried, $f(x)$ would equal 24, which is too large, so that x must be less than 6, and so on. Such smooth functions present problems in public-key systems, which depend on functions for shielding numbers.

Consider what happens, however, if modularity is added. When $f(x)$ equals, say, $4x \text{ modulo } 7$, as x increases, the value of $f(x)$ jumps around in a quite haphazard way. For example, $f(1)$ is 4, $f(2)$ is 1, $f(3)$ is 5, $f(4)$ is 2, $f(5)$ is 6 and $f(6)$ is 3. Even in such a simple case it is clear that this function that includes modularity provides far better protection for the values of x than the one that does not include it. In the case of the trapdoor knapsack system, applying modularity in the generation of the difficult knapsack vector a prevents the recovery of a' for anyone who does not know the secret transformation parameters w and m .

For anyone who does know w and m ,



Eye-level Weight Watcher

Separate digital readout puts your exact weight right before your eyes.

Your present scale gives you a number for every five pounds - and a line for the pounds in between. To make matters worse, the needle's at your feet.

Our scale puts your weight where your eyes are—in bold, red, easy-to-read numbers. It's marvelous! You step on the scale and your weight's right there. A visual record of the pounds you shed.

The readout unit mounts with either self-adhesive clips or screws. The color's soft white. The physical size of the scale unit is 10.5" x 10.5" x 2.5" and the digital readout unit is 4.75" x 3" x 1.25". Try it on a 15-day money back guarantee. Call toll free to charge it to any national credit card or send your check for \$49.95 plus \$5.50 shipping and handling to Douglas Dunhill. (Ill. residents add sales tax.) Complete with four AA batteries. Watching your weight's never been easier or as nice.

**Call Toll-Free
800-621-5554**

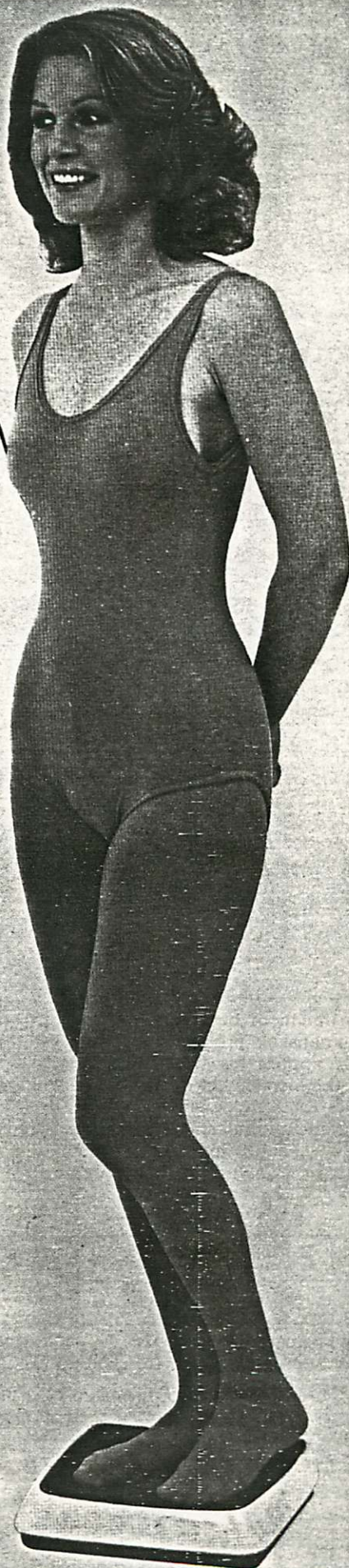
Illinois residents please call 800-972-5858
(In operation 24 hours, 7 days a week)



- Computer controlled electronic accuracy
- Automatically adjusts to zero
- 300 lb. capacity
- A fraction of the price of doctor-type pedestal scales without a digital readout

Douglas Dunhill
INC. AFFORDABLE QUALITY

Dept. 53-2578
Ten Douglas Dunhill Drive, Oak Forest, IL 60452
© Douglas Dunhill Inc. 1979



however, the conversion back into a' would not be difficult at all. In fact, with those parameters it is quite easy to convert the difficult knapsack problem involving the vector a and the transmitted ciphertext message C into an easy knapsack problem involving the vector a' and a new sum C' and then to solve (or decipher) for x . To begin with, it is a simple mathematical exercise to calculate the inverse of w modulo m , that is, the number w^{-1} that when multiplied by w modulo m gives 1. There is a fast procedure for finding inverses in modular arithmetic (based on Euclid's algorithm for finding the greatest common divisor of two numbers) that makes this calculation efficient, even in a realistic system in which w and m are on the order of 50 digits long. (Incidentally, for this purpose w and m must be chosen to be relatively prime; if they had a common factor, or divisor, there would be no multiplicative inverse of w modulo m .)

To decipher the message C , then, the receiver first calculates $C' = Cw^{-1}$ modulo m . To see what this operation accomplishes remember that C equals $a_1x_1 + \dots + a_nx_n$. In modular arithmetic, as in ordinary arithmetic, it is permissible to multiply both sides of an equation by the same quantity so that Cw^{-1} modulo m equals $a_1x_1w^{-1} + \dots + a_nx_nw^{-1}$, or $a_1w^{-1}x_1 + \dots + a_nw^{-1}x_n$ modulo m . The vector a was generated from the vector a' , however, by computing $a_i = a'_i w$ modulo m for each i . Hence $a_i w^{-1}$ equals a'_i modulo m for each i , that is, $a_1 w^{-1}$ equals a'_1 modulo m , $a_2 w^{-1}$ equals a'_2 modulo m and so on. Substituting these last results into the preceding equation, one discovers that C' , or Cw^{-1} modulo m , equals $a'_1x_1 + \dots + a'_nx_n$, or $a' \cdot x$.

In other words, calculating Cw^{-1} modulo m is all that is needed to convert the problem of deciphering C into an easy knapsack problem. The receiver simply applies his secret vector a' to solve the knapsack problem for C' and recover x . For those who do not have the secret information w and m , however, there is no easily implemented method known of transforming C into C' (or

translating the difficult vector a into the easy vector a') for efficient deciphering.

In the 10-element example I have been discussing it is easy to verify that the numbers w and m relating a and a' or C and C' are respectively 764 and 2,731, and that w^{-1} is 1,605. Notice that in this public-key system the trapdoor information w , m and a' is virtually synonymous with the secret deciphering key w^{-1} , m and a' . The same is not true of all public-key cryptosystems. (In practical cryptosystems based on the trapdoor knapsack scheme it may be desirable to introduce additional security by iterating the conversion process, so that the public and the private vectors differ by several transformations and several intermediate vectors.)

Since only the numbers w , or w^{-1} , and m and the vector a' must be kept secret, all users of the trapdoor knapsack system can employ the same public computer program for generating both their public key and their secret parameters. Utilizing a random-number generator to provide the program with a' , w and m will serve to ensure that each user's pair of keys is distinct. Similarly, a public program could be made available that would encipher messages and, when it was supplied with the secret parameters, decipher messages. Therefore no mathematical ability is required to implement the trapdoor knapsack cryptosystem. Any useful public-key system must have this same characteristic.

The second public-key system I shall describe is based on an NP problem that has an even longer and more distinguished history of resisting solution than the knapsack problem: the problem of factoring a large number, or finding all the primes that divide it evenly. (A prime number is an integer that is divisible only by 1 and itself.) This problem has been studied since the time of the ancient Greeks, and although some progress has been made with it, factoring a 200-digit number would still take the most powerful modern computer about a billion years. To give a smaller example, consider the problem of fac-

toring 29,083. Calculating by hand, it would take the better part of an hour to find the only two factors of this number: 127 and 229. It takes less than a minute, however, to verify that those factors are correct, suggesting that the problem of factoring is a good basis for the construction of a one-way function. Figuring out how to build a trapdoor into such a function presents more difficult obstacles, but they have been overcome by Rivest, Shamir and Adleman, the designers of the RSA system.

To generate a public enciphering key each user of the RSA public-key system (or rather a program run on his computer) chooses two large random prime numbers p and q . The product n of these two numbers and another random number E are placed in the public file as the user's enciphering key. To apply the key a sender first converts his message into a string of numbers, which he then breaks into blocks P_1, P_2, \dots . In this instance it is not necessary to use binary numbers, but each plaintext number P_i must be between 0 and $n - 1$. (The enciphering and deciphering functions operate modulo n and so can distinguish between numbers in this range only.) Locating the user's public key (E, n) in the directory, the sender computes for each plaintext number P_i the ciphertext number $C_i = P_i^E$ modulo n . For example, if p equals 5, q equals 11 and E equals 3, then the user's enciphering key is (3, 55), and to encipher the plaintext information $P = 2$ a sender would compute $C = 2^3 = 8$ modulo 55. (Because the numbers in this example are so small modularity does not yet play a role.)

The RSA public-key cryptosystem is based on the fact that although finding large prime numbers is computationally easy, factoring the product of two such numbers is at present computationally infeasible. (It is important to understand that because there are computationally efficient primality tests, determining whether a number is prime is much easier than factoring a number of about the same size.) To decipher a ciphertext C_1, C_2, \dots , the user employs n and a secret deciphering key D derived from the prime factors p and q of n .

To understand how the deciphering key is derived it is necessary to consider the number $(p - 1)(q - 1)$: a well-known object in number theory called Euler's totient function. This function, which is written $\phi(n)$, is defined as the number of integers between 1 and n that have no common factor with n . It is not hard to see that if n equals pq , then $\phi(n)$ equals $(p - 1)(q - 1)$. The number $\phi(n)$ is introduced here because in functions, such as the one used for enciphering in the RSA system, that are calculated modulo n , arithmetic in the exponent is carried out not modulo n but modulo $\phi(n)$. An example may make this idea clearer. Consider the expression 2^{11}

P	0	1	2	3	4	5	6	7	8	9	10	...
$C = P^3$	0	1	8	27	64	125	216	343	512	729	1000	...
$C' = P^3 \text{ modulo } 11$	0	1	8	5	9	4	7	2	6	3	10	...

MODULAR ARITHMETIC is employed in many cryptosystems to further disguise information already transformed by an enciphering function. As is shown here, the value of an integer a modulo another integer b is defined as the remainder left when a is divided by b . For example, 27 modulo 11 equals 5, because 11 goes into 27 twice with 5 left over. The usefulness of this operation is shown for a simple enciphering function $C = P^3$. As P increases, the continuous way P^3 increases makes it possible to invert the function, or determine what value of P corresponds to a particular value of C , even though there is no simple formula for expressing P as the cube root of C . More precisely, a value of P that gives too small a value of C is itself too small, whereas value of P that gives too large value of C is itself too large. When modularity is added, however, so that C' equals P^3 modulo 11, values of function are thrown into disarray. As P increases, C' changes in a quite discontinuous way, effectively shielding P .

In Saronno, all we think about is love.



That's all we've been thinking about for 450 years. Because this is where the drink of love began. With Amaretto di Saronno. If what you're drinking doesn't come from Saronno, how do you know it's love?

Amaretto di Saronno. The Original.

Liqueur 56 proof. Imported by Foreign Vintages, Inc., Jericho, New York. © 1978

SIGMA

75-250MM ONE-TOUCH ZOOM:

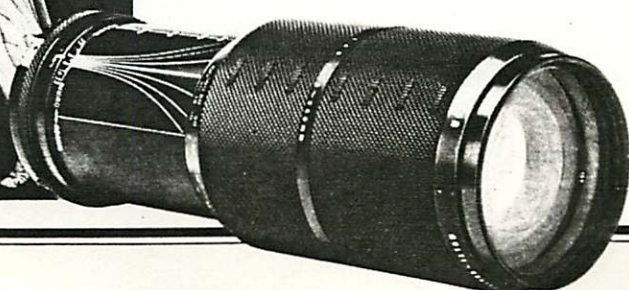
Zoom. Focus. Come closer. All with a single control.



Sigma's compact new 75-250mm zoom runs rings around ordinary zooms. You zoom, focus, even take close-ups *all with the same super-fast control!* No fumbling for separate rings and levers, ever. And its 176 focal lengths bring the action up to five times closer than a 'normal' lens.

Like all Sigma lenses from wide angle to super mirror telephoto, its computer derived, multicoated optics give razor-sharp images far or near... at a price far less than you'd expect. Complete with case and lens hood, for all popular 35mm SLR cameras. So zoom down to your Sigma dealer today. Or write for LitPak P84.

UNITRON Unitron Instruments, Inc., Woodbury, NY 11797. A subsidiary of Ehrenreich Photo-Optical Industries, Inc.

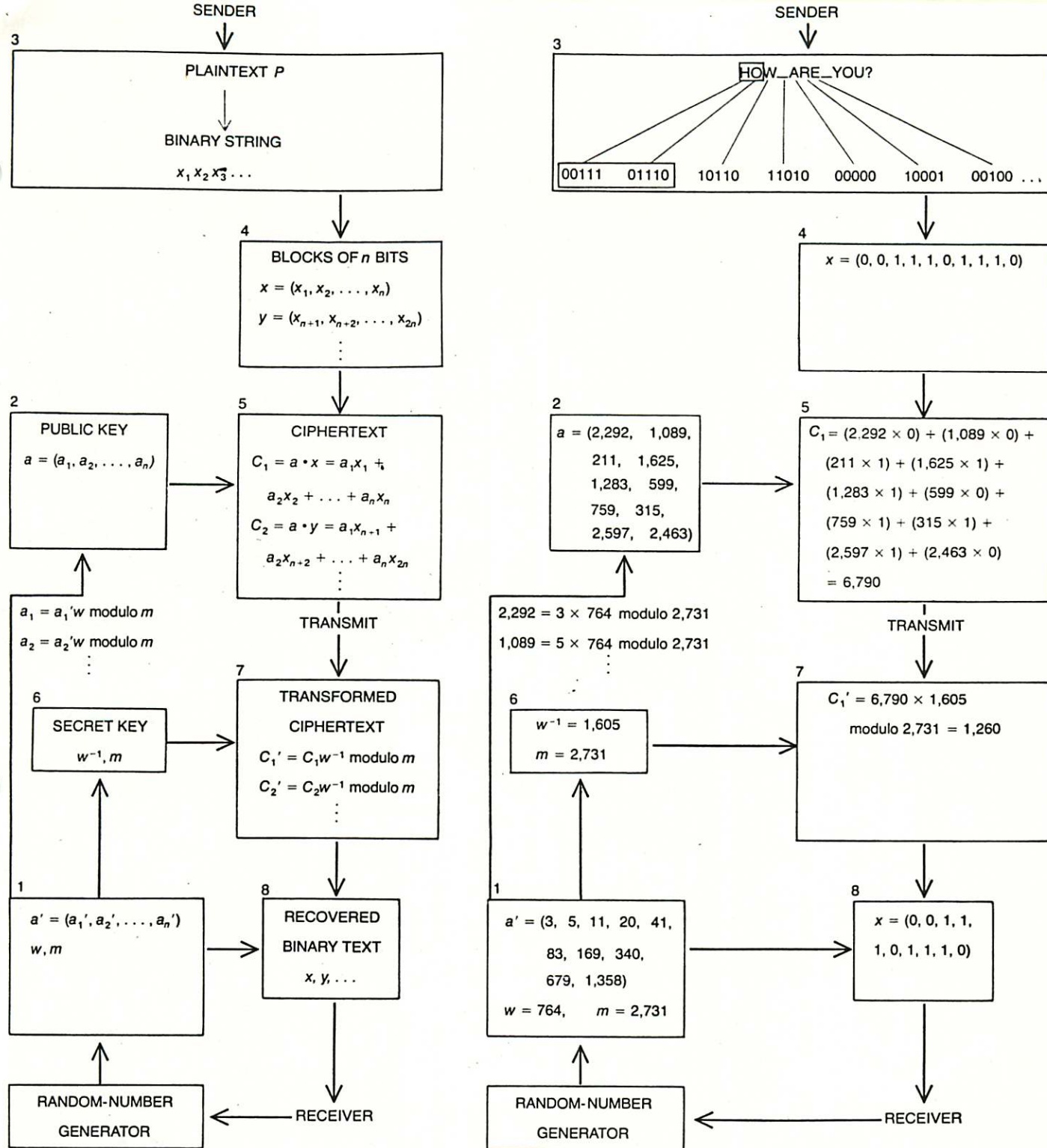


modulo 10. Since 2^{11} is equal to 2,048 and dividing 10 into 2,048 leaves a remainder of 8, the expression is equal to 8. Note that calculating by first reducing the exponent modulo 10 does not give the correct answer, since 11 modulo 10 equals 1, and 2^1 equals 2. On the other hand, 10 equals 2×5 , and so $\phi(10)$ equals $(2-1)(5-1)$, or 4. Since 11 modulo 4 equals 3, calculating 2^{11} modulo 10 by first reducing the exponent modulo 4 gives the correct answer: 2^3 , or 8.

Now, the properties of $\phi(n)$ guarantee that there is always a multiplicative inverse D of E modulo $\phi(n)$, that is, ED modulo $(p-1)(q-1)$ is equal to 1. In fact, there is always a fast, computationally easy method for deriving D . (It is not hard to see that in the example discussed above, where p equals 5, q equals 11 and E equals 3, $(p-1)(q-1)$ equals 40 and D equals 27, because 3×27 is one more than 2×40 .) This inverse D is the secret deciphering key for the RSA system. To decipher a ciphertext the receiver computes C_i^D modulo n for each ciphertext number C_i . C_i equals P_i^E modulo n , so that C_i^D modulo n equals $(P_i^E)^D$, or P_i^{ED} modulo n . Because arithmetic in the exponent is performed modulo $\phi(n)$ and ED modulo $\phi(n)$ equals 1, P_i^{ED} modulo n equals P_i^1 , or P_i . In other words, raising the ciphertext to the D th power and reducing modulo n recovers the plaintext.

Hence in the RSA cryptosystem modularity plays a dual role, not only blocking the recovery of the secret deciphering key D from the public enciphering key (E, n) but also, by its presence in the enciphering algorithm, preventing a direct recovery of the plaintext from the ciphertext. The difficulty of computing D from the public information (E, n) depends on the difficulty of factoring n , or of deriving p and q from n . Once again the example I have given is too small to provide real secrecy, but since factoring large numbers is a very difficult problem, the difficulty of breaking the cipher blows up rapidly as n increases. When p and q are chosen so that n is about 200 digits long, it appears to be computationally infeasible for anyone but the intended receiver to decipher the message.

Just as the deciphering procedure (without the trapdoor information) must be computationally infeasible, the public enciphering procedure and secret deciphering procedure must be computationally efficient. At first the implementation of the RSA system appears to present some practical problems in this area. Consider the simple example in which the plaintext number $P = 2$ was transformed into the ciphertext number $C = 8$. To apply the deciphering algorithm $P = C^D$ modulo n it is necessary to calculate 8^{27} modulo 55 (which does indeed equal 2). Multiplying 8 by itself 27 times is, however, a cumbersome process involving large numbers and a great



FLOW OF INFORMATION in the trapdoor knapsack cryptosystem is shown at the left. The corresponding transformations of the first block of plaintext HO are shown at the right. In this public-key system based on the knapsack problem each receiver (by means of a random-number generator) selects a secret vector $a' = (a'_1, \dots, a'_n)$ with the property that each element is greater than the sum of the preceding elements and also selects two large random numbers w and m with no common factors (1). The numbers w and m are the trapdoor parameters for converting the secret "easy" vector a' into a "difficult" public vector $a = (a_1, \dots, a_n)$ by means of the equations $a_1 = a'_1 w \text{ modulo } m$, $a_2 = a'_2 w \text{ modulo } m$ and so on. The difficult vector a is transmitted to the sender over an insecure channel or is listed in a public directory as the receiver's enciphering key (2). To encipher a plaintext P a sender begins by converting it into a binary string according to, say, the five-bit binary alphabet given at the top of the illustration on page 146 (3). The sender then looks up the receiver's public key and breaks the string into blocks of n binary digits (4). For example, in the system shown at the right (in which the numbers are

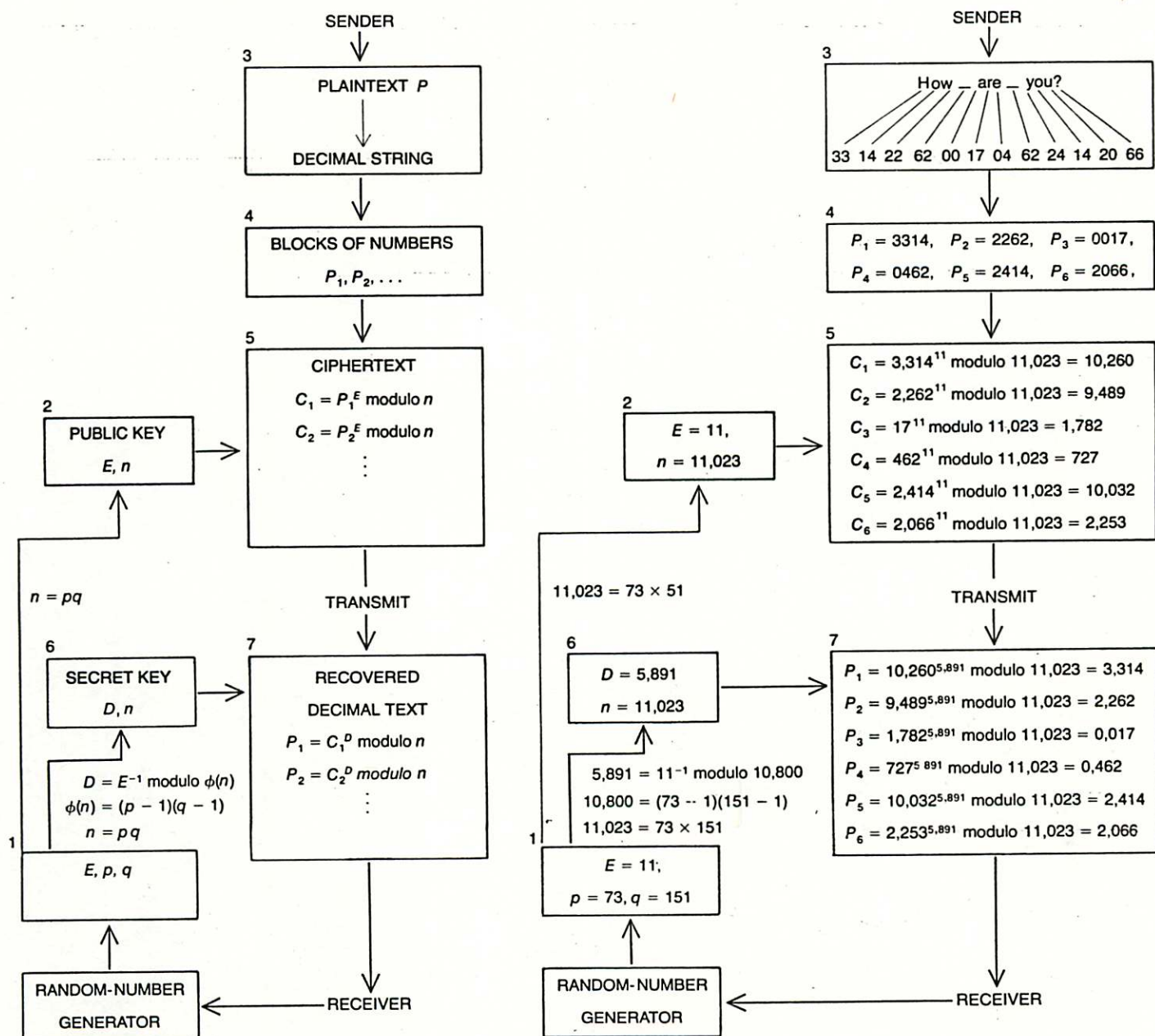
far too small to provide real secrecy) there are 10 elements in the public vector, and so the binary string is divided into blocks of 10 bits each. Every block is enciphered by forming its dot product with a (5), that is, if the first block is $x = (x_1, \dots, x_n)$, the first ciphertext number C_1 equals $a \cdot x$, or $a_1 x_1 + \dots + a_n x_n$, and so on. The ciphertext numbers are transmitted to the receiver over an insecure channel. The receiver recovers, say, x by calculating first w^{-1} (6) and then $C_1' = C_1 w^{-1} \text{ modulo } m$ (7). (The number w^{-1} is the inverse of w modulo m , the number that when multiplied by w gives 1 modulo m .) Remember that a_1 equals $a'_1 w \text{ modulo } m$, a_2 equals $a'_2 w \text{ modulo } m$ and so on, and therefore $a_1 w^{-1}$ equals $a'_1 \text{ modulo } m$, $a_2 w^{-1}$ equals $a'_2 \text{ modulo } m$ and so on. Hence C_1' , or $C_1 w^{-1} \text{ modulo } m$, which equals $a_1 x_1 w^{-1} + \dots + a_n x_n w^{-1}$, or $a'_1 x_1 + \dots + a'_n x_n \text{ modulo } m$, equals $a' \cdot x$. In other words, C_1' equals $a' \cdot x$, and the difficult knapsack problem of recovering x from C_1 and a has been converted back into the easy problem of recovering x from C_1' and a' . Only receiver, who possesses trapdoor information w and m and knows secret vector a' , can effect transformation and recover x (8).

many computational steps. In more realistic RSA systems, where D would be a 200-digit number, this procedure would be impossible to carry out, even on a very powerful computer.

Fortunately there is a much faster method for calculating functions of this kind. First the binary expansion of the exponent (the expression of the exponent as a sum of powers of 2) is utilized to break up the function into a product

of smaller factors; for example, 27 equals $1 + 2 + 8 + 16$, and so 8^{27} equals $8 \times 8^2 \times 8^8 \times 8^{16}$. Now, by calculating the smaller factors first and then taking their product, the number of operations required can be limited. For example, 8^2 modulo 55 can be evaluated with only one modular multiplication (an ordinary multiplication followed by an ordinary division), since 8×8 , or 64, modulo 55 equals 9. Then 8^4 , which

is equal to $8^2 \times 8^2$, can be evaluated with an additional modular operation, $9 \times 9 = 81 = 26$ modulo 55, and so on. (Substituting the value of 8^2 modulo 55, namely 9, into the larger factors prevents the size of the numbers involved in the computation from blowing up.) Hence only seven modular multiplications are needed to calculate 8^{27} : four to evaluate 8^2 , 8^4 , 8^8 and 8^{16} and three more to multiply 8 times 8^2 times 8^8



RSA PUBLIC-KEY CRYPTOSYSTEM is based on the problem of factoring a large number, or finding all the prime numbers that divide it evenly. (A prime number is an integer that is divisible only by 1 and itself.) As is shown at the left, each receiver in the RSA system generates two large random prime numbers p and q , which serve as his secret trapdoor parameters, and a large random number E (1). There are computationally efficient tests for identifying primes such as p and q , but when these numbers are sufficiently large, it is computationally infeasible to derive them from their product pq , or n . The receiver lists E and n as his public deciphering key (2). To encipher a plaintext P the sender first converts it into a string of numbers, using the decimal alphabet shown at the bottom of the illustration on page 146 (3), and then breaks the string into blocks of equal length P_1, P_2, \dots , so that each number P_i in this series is less than n (4). Each block is convert-

ed into a ciphertext number by raising it to the E th power and then reducing modulo n , that is, C_1 equals $P_1^E \text{ modulo } n$, C_2 equals $P_2^E \text{ modulo } n$ and so on (5). The ciphertext numbers C_1, C_2, \dots are transmitted over an insecure channel. Arithmetic in the exponent of a function that is calculated modulo n must be carried out modulo $\phi(n)$, where $\phi(n)$ equals $(p-1)(q-1)$, and so the receiver utilizes p and q to determine $\phi(n) = (p-1)(q-1)$ and then $D = E^{-1} \text{ modulo } \phi(n)$, which serves as his secret deciphering key (6). To convert C_1, C_2, \dots back into plaintext numbers the receiver raises each one to the D th power and reduces it modulo n (7). Because $C_i^D \text{ modulo } n$ equals $(P_i^E)^D \text{ modulo } n$, or $P_i^{ED} \text{ modulo } n$ and $ED \text{ modulo } \phi(n)$ equals 1, $P_i^{ED} \text{ modulo } n$ equals $P_i \text{ modulo } n$. In other words, this operation inverts the enciphering transformation, recovering the plaintext number blocks P_1, P_2, \dots . In the example shown at right the values of E, p and q are too small to provide real security.

times 8^{16} . Even when D is a 200-digit number, this method results in a deciphering procedure that is quite efficient, requiring at most 1,330 modular multiplications rather than the 10^{200} operations necessary in the straightforward approach.

The RSA system is a public-key cryptosystem that allows the direct generation of a digital signature: a number that can be appended to a ciphertext message to solve the problems of authentication mentioned above. To be of real service such a signature must be easy for the sender to generate and for the receiver to check but must be computationally infeasible for a third party or the receiver himself to generate. Of the various methods for generating digital signatures the simplest involves exploiting the inverse relation of the public enciphering and secret deciphering keys by reversing their roles. For example, in the RSA system the sender can utilize his own secret deciphering key D as a signing key, to compute the signature $S_i = P_i^D$ modulo n for each P_i in the series of plaintext numbers P_1, P_2, \dots that represent a message to be transmitted. (Remember that each P_i is chosen to be between 0 and $n - 1$.) Once the signatures S_1, S_2, \dots have been generated the sender enciphers each signed message block (P_i, S_i) , using the receiver's public enciphering key. This second operation has nothing to do with signature generation; it simply ensures the privacy of the communication.

The receiver uses his own secret key to recover the signed message block (P_i, S_i) , and then he looks up the sender's public key (E, n) and computes S_i^E modulo n for each i . D and E effect inverse operations regardless of the order of their application, and so since S_i equals P_i^D modulo n , S_i^E modulo n should be equal to P_i . If that is the case, the receiver can be sure that the message comes from the apparent sender and that it has not been tampered with. Since the digital signature depends on both the sender and the message sent, it offers a level of security different from that of a written signature, which is the same for all messages. With the digital signature neither the receiver nor a third party can alter the message without destroying the validity of the signature. (When the message to be sent is long, rather than signing each submessage separately it may be desirable to compress the message and calculate a single signature S ; that compression can be effected in such a way that S still depends on the entire message P .)

The traditional difficulties of solving the knapsack and factoring problems can be taken as an encouraging sign that the public-key cryptosystems based on these problems are in practical terms secure. A past history of intracta-

bility cannot, however, be considered a proof that a system is secure. It is always possible, if unlikely, that at some time in the future computationally efficient general methods for solving these problems will be found. An even greater hazard is that a method will be discovered for breaking one of the cryptosystems without solving the corresponding general problem. For example, it is possible that although solving most regular knapsack problems is computationally infeasible, there is an easy way to solve the much smaller set of trapdoor knapsack problems. Similarly, it may be possible to recover the plaintext enciphered by the RSA technique without finding the factors of n . (Michael O. Rabin of the Hebrew University of Jerusalem has recently shown, however, that in the case where the enciphering exponent E is 2, the security of the RSA system is not simply dependent on the difficulty of factoring n but is actually equivalent to it. This finding constitutes an important first step toward the goal of developing provably secure systems.)

Cryptography has not yet advanced to the stage where it can prove the computational security of even a conventional system or a one-way function. Hence it is not surprising that there is no way to establish the security of the public-key systems, which are based on the more complex trapdoor one-way functions. It is hoped, however, that over the next decade or two complexity theory will advance to the point where such proofs can be formulated. Some progress has been made in this direction through the study of a special subset of the NP problems.

Remember that the NP problems are ideal candidates for one-way functions because finding a solution to them is computationally difficult but checking a proposed solution is computationally easy. Some of these problems such as the knapsack problem (but not factoring) belong to the subset of the NP problems that is called NP-complete. The NP-complete problems have the added property that if any one of them had an easily implemented method for finding general solutions, then all the NP problems would. Now, all cryptanalytic problems—problems of breaking cryptosystems—are in the class NP, since it is always easy to check the validity of a proposed key. Therefore if any NP-complete problem can be solved rapidly, it follows that all cryptographic systems can be broken easily. Roughly speaking, then, if the security of a cryptosystem could be shown to be equivalent in difficulty to an NP-complete problem, it would be as secure as any cryptographic system can be.

One flaw in this type of evaluation is that complexity theory deals with the "worst case" computational difficulty of solving a problem, whereas cryptography is concerned with the average or

typical difficulty of solving a problem. For example, in current complexity theory a problem whose solution requires $10^{1,000}$ operations 1 percent of the time but only 100 operations 99 percent of the time is considered to be difficult. Obviously a cryptosystem that can be broken 99 percent of the time is worthless. Workers in complexity theory are aware of this shortcoming and are currently developing more suitable measures of computational difficulty.

Although factoring is not an NP-complete problem, it has through the years largely resisted the attack of some of the best mathematical minds. That is why Rabin's proof, which establishes an equivalence between the difficulty of factoring and breaking an RSA scheme, is an extremely important result. Until such time as the security of proposed cryptosystems can be formally evaluated, however, it is a worthwhile (and intellectually challenging) exercise to try to break them.

In electronic communications systems, as in any new technology, there is a potential for misuse. For example, the danger of foreign or domestic intelligence organizations spying on American citizens who rely on these systems is a real one. It has recently been revealed that U.S. microwave telephone traffic is being monitored in at least one foreign embassy in Washington. In the late 1960's the U.S. Government's "Operation Shamrock" intercepted international Telex communications to and from "targeted" individuals, including anti-war activists. If such excesses are to be limited, both legal and technical safeguards are needed.

There is always a trade-off between the rights of citizens to privacy and the desire of government intelligence agencies to limit the availability of secure cryptosystems. A conflict in this area has recently arisen concerning the Federal Data Encryption Standard, a conventional cryptosystem issued by the National Bureau of Standards for non-military encryption purposes. The National Security Agency convinced the International Business Machines Corporation, the company that designed the standard, to reduce the key size to 56 bits. Although there is controversy surrounding the issue, I believe the reduction in key size was meant to weaken the standard so that if it were ever employed by a foreign organization, it could be broken by the National Security Agency. Issues similar to this one will certainly arise as the new public-key systems become commercial realities. It is to be hoped that these issues will be decided by an open discussion of the relative needs of the intelligence community and the citizenry rather than, as appears to have been the case, by the unilateral decision of the intelligence community that its needs take precedence.

THE AUTHORS

RAYMOND DEVORET ("Bacterial Tests for Potential Carcinogens") is on the research staff of the Enzymology Laboratory at the Centre National de la Recherche Scientifique (C.N.R.S.) at Gif-sur-Yvette in France. He obtained his medical degree at the University of Paris Medical School and then worked part-time as a physician specializing in occupational radiation hazards. At the same time he took a degree in physics from the University of Paris and studied the effects of alpha rays on the bacterium *Escherichia coli* at the Radium Institute. In 1961 he was appointed to a full-time position in the C.N.R.S. Since then Devoret's research has been devoted to elucidating the mechanism of induction of dormant viruses in bacteria and the mechanism of chemical carcinogenesis.

HAROLD P. FURTH ("Progress toward a Tokamak Fusion Reactor") is professor of astrophysical sciences at Princeton University and associate director of the Princeton Plasma Physics Laboratory. He did graduate work in solid-state and high-energy physics at Harvard University, receiving his Ph.D. in 1960. He then joined the controlled-fusion research effort at the Lawrence Livermore Laboratory, becoming head of a group studying the toroidal magnetic-confinement approach to a fusion reactor. In 1967 he moved to Princeton, where he has initiated a number of major toroidal confinement experiments and done extensive research on the theory of plasma instabilities. His best-known publication is a poem in *The New Yorker* that describes an encounter between matter and antimatter, exemplified by the meeting of Dr. Edward Teller and Dr. Edward Anti-Teller. Furth would like to acknowledge the assistance of Robert Goldston in the preparation of his article.

ANDREW M. T. MOORE ("A Pre-Neolithic Farmers' Village on the Euphrates") is Wainwright Fellow in Near Eastern Archaeology at the University of Oxford. He studied modern history at Oxford as an undergraduate and then was trained in archaeology at the Institute of Archaeology of the University of London. In 1969 he was given a scholarship at the British School of Archaeology in Jerusalem, and for the next two years he lived in the Near East studying museum collections and visiting ancient sites. In 1973 he was awarded a Wainwright fellowship at Oxford; he obtained his Ph.D. last year. Moore has done extensive field work in the Near East: he took part in excavations at Jerusalem directed by the late Dame Kathleen Kenyon in 1966 and those at Knossos in Crete directed by J. D. Evans in

1969. From 1972 to 1973 Moore was in charge of the major excavation at Tell Abu Hureyra in Syria that is described in his article. This fall Moore will be a visiting professor in Old World prehistory at the University of Arizona.

THOMAS F. GOREAU, NORA I. GOREAU and THOMAS J. GOREAU ("Corals and Coral Reefs") have done pioneering research in coral-reef biology and ecology in Jamaica. Thomas F. Goreau died in 1970. The son of the photographers Fritz and Grete Goro, he was born in Germany, which his parents left in the early 1930's for France and soon thereafter the U.S. After his graduation from Clark University he did graduate work in ecology at Yale University under G. Evelyn Hutchinson. Goreau first became interested in coral-reef ecology while serving in the 1947 Bikini Scientific Resurvey, although the first opportunity to pursue this line of research did not arise until 1951, when he moved to Jamaica to join the faculty of physiology in the medical school of the University of the West Indies. On receiving his Ph.D. in 1956 he instituted, under the auspices of the New York Zoological Society, a long-term research project on Jamaican coral reefs that continued until 1967. In March, 1970, shortly before his death, he opened a new marine laboratory at Discovery Bay on the northern coast of Jamaica. Goreau's widow, Nora Goreau, is currently senior research fellow in marine biology at the University of the West Indies. Born in Panama, she was educated at the University of Panama Law School, the University of Iowa and DePaul University. She then did graduate work in neurophysiology at the University of Chicago. Thomas J. Goreau, their eldest son, is a Ph.D. candidate in geology at Harvard University. He did his undergraduate work in physics and astronomy at the Massachusetts Institute of Technology and obtained his master's degree at the California Institute of Technology in 1972. He then studied biology and oceanography at Yale, the Woods Hole Oceanographic Institution and Harvard.

WILLIAM HERBST and GEORGE E. ASSOUSA ("Supernovas and Star Formation") have collaborated on studies of star formation with optical and radio telescopes. Herbst is assistant professor of astronomy at Wesleyan University. He did his undergraduate work in astrophysics at Princeton University and received his Ph.D. in astronomy from the University of Toronto in 1974. After a postdoctoral fellowship at York University he was a fellow for two years of the Carnegie Institution of Washing-

ton. Herbst joined the Wesleyan faculty in 1978. Assousa is research professor of astrophysics in the Department of Terrestrial Magnetism of the Carnegie Institution. A Palestinian Arab, he was born in Jerusalem in 1936 and attended the American Friends School at Ramallah in Jordan. He then came to the U.S. to finish his education, obtaining his bachelor's degree at Earlham College, his master's degree at Columbia University and his Ph.D. in experimental nuclear physics from Florida State University in 1968. That year he was appointed a Carnegie fellow; two years later he joined the staff of the Carnegie Institution. Assousa directs the Foundation for Arab-Israeli Reconciliation (FAIR), which seeks to improve communication and cooperation between Arab and Israeli professionals and scholars.

MARTIN E. HELLMAN ("The Mathematics of Public-Key Cryptography") is associate professor of electrical engineering at Stanford University. He received his B.E. at New York University and his Ph.D. from Stanford in 1969. After teaching for two years at the Massachusetts Institute of Technology he joined the Stanford faculty. He is best known for his invention of public-key cryptography in collaboration with his students Whitfield Diffie and Ralph Merkle; he has also done research on information theory, error-control coding and statistics. Hellman wishes to acknowledge the National Science Foundation's support of his work.

KENNETH G. WILSON ("Problems in Physics with Many Scales of Length") is James A. Weeks professor of physical science at Cornell University. He obtained his bachelor's degree at Harvard University and his Ph.D. from the California Institute of Technology in 1961. The renormalization-group theory, his principal contribution to physics, resulted from his attempt to understand quantum field theory. He is currently applying improved computer technology to extend the capabilities of the renormalization-group approach. His pastimes include international folk dancing and detective stories. Wilson also has to his credit a 4:17 mile.

DAVID CREWS ("The Hormonal Control of Behavior in a Lizard") is associate professor of biology and psychology at Harvard University and an associate of the Museum of Comparative Zoology. He did his undergraduate work at the University of Maryland and received his Ph.D. in animal behavior from Rutgers University. He then did postdoctoral research at the University of California at Berkeley and at Harvard, where he was appointed to the faculty in 1976. Crews's interest in reptiles and their behavior goes back to his childhood in Florida.

BIBLIOGRAPHY

Readers interested in further explanation of the subjects covered by the articles in this issue may find the following lists of publications helpful.

MATHEMATICAL GAMES

CONVERGENCE ON THE ARGAND DIAGRAM. L. W. H. Hull in *The Mathematical Gazette*, Vol. 43, No. 345, pages 205-207; October, 1959.

COMPLEX NUMBERS IN GEOMETRY. I. M. Yaglom. Academic Press, 1968.

THE HISTORICAL DEVELOPMENT OF COMPLEX NUMBERS. D. R. Green in *The Mathematical Gazette*, Vol. 60, No. 412, pages 99-107; June, 1976.

BACTERIAL TESTS FOR POTENTIAL CARCINOGENS

METHODS FOR DETECTING CARCINOGENS AND MUTAGENS WITH THE SALMONELLA/MAMMALIAN-MICROSOME MUTAGENICITY TEST. Bruce N. Ames, Joyce McCann and Edith Yamasaki in *Mutation Research*, Vol. 31, No. 6, pages 347-364; December, 1975.

PROPHAGE λ INDUCTION IN *ESCHERICHIA COLI* K12 ENV A UVRB: A HIGHLY SENSITIVE TEST FOR POTENTIAL CARCINOGENS. Patrice Moreau, Adriana Bailone and Raymond Devoret in *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 73, No. 10, pages 3700-3704; October, 1976.

ULTRAVIOLET MUTAGENESIS AND INDUCIBLE DNA REPAIR IN *ESCHERICHIA COLI*. Evelyn M. Witkin in *Bacteriological Reviews*, Vol. 40, No. 4, pages 869-907; December, 1976.

PROGRESS TOWARD A TOKAMAK FUSION REACTOR

TOKAMAK DEVICES. L. A. Artsimovich in *Nuclear Fusion*, Vol. 12, No. 2, pages 215-252; March, 1972.

TOKAMAK RESEARCH. H. P. Furth in *Nuclear Fusion*, Vol. 15, No. 3, pages 487-534; June, 1975.

A PHYSICIST'S ABC ON PLASMA. L. A. Artsimovich. Mir Publishers, 1978.

RECENT PROGRESS IN TOKAMAK EXPERIMENTS. Masanori Murakami and Harold P. Eubank in *Physics Today*, Vol. 32, No. 5, pages 25-32; May, 1979.

A PRE-NEOLITHIC FARMERS' VILLAGE ON THE EUPHRATES

PAPERS IN ECONOMIC PREHISTORY. Edited by E. S. Higgs. Cambridge University Press, 1972.

FARMING IN PREHISTORY. Barbara Bender. St. Martin's Press, 1975.

THE EXCAVATION OF TELL ABU HUREYRA IN SYRIA: A PRELIMINARY REPORT. A. M. T. Moore in *Proceedings of the Prehistoric Society*, Vol. 41, pages 50-77; December, 1975.

THE NEOLITHIC OF THE NEAR EAST. James Mellaart. Charles Scribner's Sons, Inc., 1976.

CORALS AND CORAL REEFS

PROBLEMS OF GROWTH AND CALCIUM DEPOSITION IN REEF CORALS. T. Goreau in *Endeavor*, Vol. 20, No. 77, pages 32-39; January, 1961.

THE BIOLOGY OF CORAL REEFS. C. M. Yonge in *Advances in Marine Biology*, Vol. 1, edited by F. S. Russel. Academic Press, 1963.

REEF CORALS: AUTOTROPHS OR HETEROTROPHS? Thomas F. Goreau, Nora I. Goreau and C. M. Yonge in *The Biological Bulletin*, Vol. 141, No. 2, pages 247-260; October, 1971.

CORAL REEFS AND MOLLUSCS. C. M. Yonge in *The Royal Society of Edinburgh Transactions*, Vol. 69, No. 7, pages 147-166; 1974.

SUPERNOVAS AND STAR FORMATION

SUPERNOVAE. D. N. Schramm. D. Reidel Publishing Co., 1977.

DARK NEBULAE, GLOBULES, AND PROTOSTARS. Bart J. Bok in *Publications of the Astronomical Society of the Pacific*, Vol. 89, No. 531, pages 597-611; October, 1977.

OBSERVATIONAL EVIDENCE FOR SUPERNOVA-INDUCED STAR FORMATION: CANIS MAJOR R1. William Herbst and George E. Assousa in *The Astrophysical Journal*, Vol. 217, No. 2, Part 1, pages 473-487; October 15, 1977.

PROTOSTARS AND PLANETS. Edited by T. Gehrels. University of Arizona Press, 1978.

THE MATHEMATICS OF PUBLIC-KEY CRYPTOGRAPHY

THE CODEBREAKERS: THE STORY OF SECRET WRITING. David Kahn. The Macmillan Company, 1968.

NEW DIRECTIONS IN CRYPTOGRAPHY. Whitfield Diffie and Martin E. Hellman in *IEEE Transactions on Information Theory*, Vol. IT-22, No. 6, pages 644-654; November, 1976.

A METHOD FOR OBTAINING DIGITAL SIGNATURES AND PUBLIC-KEY CRYPTOSYSTEMS. R. L. Rivest, A. Shamir and A. Adleman in *Communications of the ACM*, Vol. 21, No. 2, pages 120-126; February, 1978.

PRIVACY AND AUTHENTICATION: AN INTRODUCTION TO CRYPTOGRAPHY.

Whitfield Diffie and Martin E. Hellman in *Proceedings of the IEEE*, Vol. 67, No. 3, pages 397-427; March, 1979.

PROBLEMS IN PHYSICS WITH MANY SCALES OF LENGTH

THE RENORMALIZATION GROUP AND BLOCK SPINS: THE BOLTZMANN MEDAL ADDRESS. Kenneth G. Wilson. Publishing House of the Hungarian Academy of Sciences.

CRITICAL PHENOMENA IN 3.99 DIMENSIONS. K. G. Wilson in *Physica*, Vol. 73, No. 1, pages 119-128; April 1, 1974.

THE RENORMALIZATION GROUP IN THE THEORY OF CRITICAL BEHAVIOR. Michael E. Fisher in *Reviews of Modern Physics*, Vol. 46, No. 4, pages 597-616; October, 1974.

RENORMALIZATION GROUP METHODS. Kenneth G. Wilson in *Advances in Mathematics*, Vol. 16, No. 2, pages 170-186; May, 1975.

THE RENORMALIZATION GROUP: CRITICAL PHENOMENA AND THE KONDO PROBLEM. Kenneth G. Wilson in *Reviews of Modern Physics*, Vol. 47, No. 4, pages 773-840; October, 1975.

INTRODUCTION TO THE RENORMALIZATION GROUP AND TO CRITICAL PHENOMENA. Pierre Pfeuty and Gérard Toulouse. John Wiley & Sons, 1977.

THE HORMONAL CONTROL OF BEHAVIOR IN A LIZARD

PSYCHOBIOLOGY OF REPTILIAN REPRODUCTION. David Crews in *Science*, Vol. 189, No. 4208, pages 1059-1065; September 26, 1975.

BIOLOGICAL DETERMINANTS OF SEXUAL BEHAVIOR. Edited by John B. Hutchinson. John Wiley & Sons, Inc., 1977.

THE PARENTAL BEHAVIOR OF RING DOVES. Rae Silver in *American Scientist*, Vol. 66, No. 2, pages 209-215; March-April, 1978.

ENDOCRINE CONTROL OF REPTILIAN REPRODUCTIVE BEHAVIOR. D. Crews in *Endocrine Control of Sexual Behavior*, edited by Carlos Beyer. Raven Press, 1979.

THE AMATEUR SCIENTIST

WATER BELLS. F. L. Hopwood in *The Proceedings of the Physical Society*, Section B, Vol. 65, Part 1, No. 385B, pages 2-5; January 1, 1952.

CAPILLARY ACTION IN SMALL JETS IMPINGING ON LIQUID SURFACES. John H. Lienhard in *Transactions of the American Society of Mechanical Engineers, Series D, Journal of Basic Engineering*, Vol. 90, pages 137-138; March, 1968.

THE BREAK-UP OF AXISYMMETRIC LIQUID SHEETS. J. C. P. Huang in *Journal of Fluid Mechanics*, Vol. 43, Part 2, pages 305-319; August 28, 1970.

by a third party. In other words, these systems make it possible to dispense with the transporting of signed documents and to depend exclusively on the electronic transmission of information.

If an eavesdropper had unlimited computing resources, he could break a public-key system and recover a plaintext. The enciphering operation E is public and the number of possible plaintexts is immense but finite, and so E could be applied to each plaintext until the intercepted ciphertext was reproduced. Since such an attack requires an impossibly large amount of computing time, however, the public-key systems can still be computationally secure. There are also similar techniques for deriving the secret deciphering key D from the public enciphering key E , but once again the computational infeasibility of implementing those algorithms provides the systems with practical security. To put it another way, the systems are based on what are called trapdoor one-way functions. A one-way function is an easily computed function for which it is computationally infeasible to compute the inverse function. A trapdoor one-way function is an easily computed function for which it is computationally infeasible to compute the inverse function unless certain specific information that was employed in the design of the function is known. Hence like a trapdoor in the floor of a motion-picture haunted house, such functions are easy to go through in one direction, but unless one possesses the special

trapdoor information (analogous in the haunted house to which brick to pull or which panel to push) the reverse process takes an impossibly long time.

The search for trapdoor one-way functions on which to base public-key cryptosystems led naturally to the class of problems that complexity theory has identified as nondeterministic, polynomial-time problems, or NP problems. For the purposes of these cryptosystems the most important property of the NP problems is that at present all the algorithms that are known for finding general solutions to them call for rapidly increasing amounts of time, although a proposed solution can be quickly checked. In other words, as the size n of such a problem increases, the number of computational steps required to solve the problem increases in proportion to, say, an exponential function of n such as 2^n , whereas the number of steps required to check a possible solution increases in proportion to a polynomial function of n such as n^2 . Exponential functions increase far more rapidly than polynomial ones, so that a method of solution that requires exponentially increasing amounts of computer time is impossible to implement for even moderate-size problems. For mathematicians concerned with cryptography the appeal of the NP problems resides in the fact that although it might take someone billions of years to find a solution to such a problem, once he found it he could convince the rest of the world of

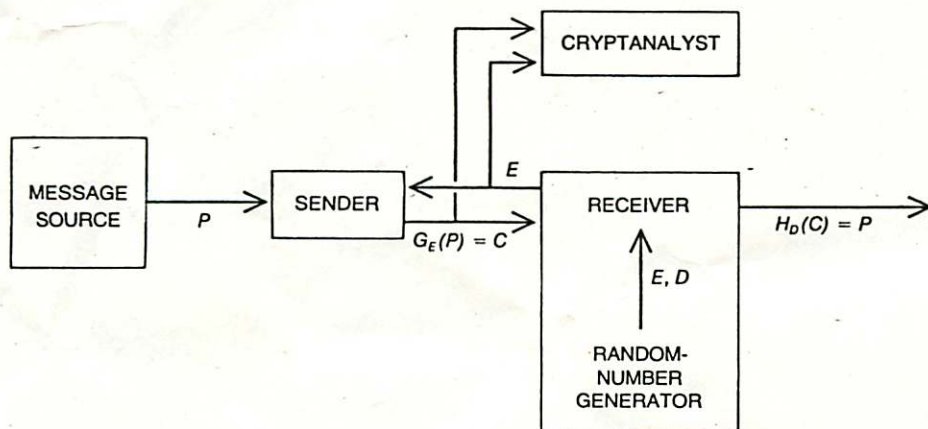
its validity in seconds. As a result these problems lend themselves readily to the construction of one-way functions. And for the NP problems on which public-key cryptosystems have been based it has been possible to build trapdoors into the functions as well.

I shall describe here two public-key cryptosystems based on NP problems: the trapdoor knapsack system, developed by Merkle and me, and the RSA system, developed by Ronald Rivest, Adi Shamir and Leonard Adleman at the Massachusetts Institute of Technology. The first of these cryptosystems is based on a well-known NP problem called the knapsack or subset sum problem: Given a knapsack of length C and a set of n rods all of the same diameter as the knapsack but of lengths a_1, a_2, \dots, a_n , find a subset of the rods that completely fills the knapsack. To put it another way, given a set of numbers a_1, \dots, a_n and a sum C , determine which of the numbers add up to C .

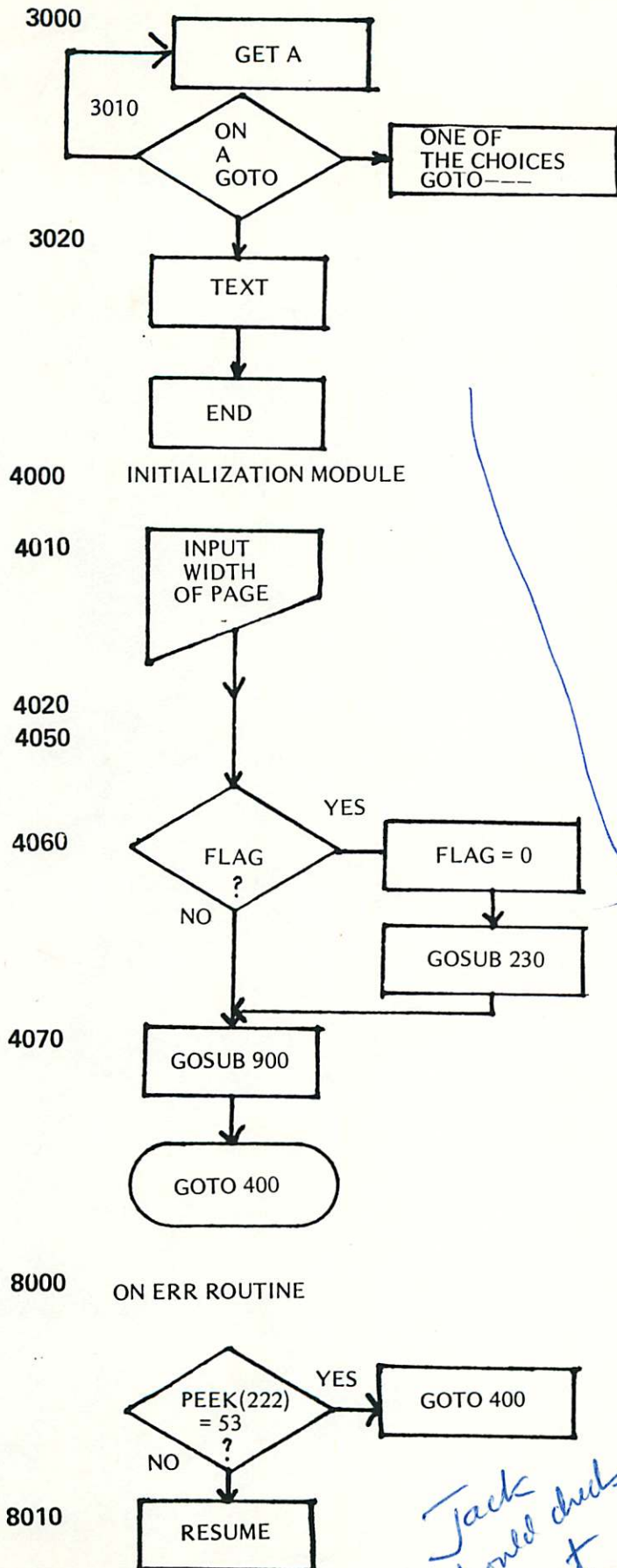
The public-key cryptosystem based on this problem operates as follows. The sender begins by converting his message into a string of binary numbers. For example, five bits (binary digits) might be allocated for each letter, number or punctuation mark in the plaintext, providing an alphabet of 2^5 , or 32, characters: A = 00000, B = 00001, C = 00010 and so on. Once the message is in binary form the sender consults a public directory of enciphering keys, which lists an ordered set of n numbers $a = (a_1, a_2, \dots, a_n)$ for each user of the system. This set is called the user's trapdoor knapsack vector.

In mathematics an ordered set of n numbers is called an n -dimensional vector, and the "dot," or scalar, product of any two vectors of the same dimension is defined as follows: for vectors $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$ the dot product $a \cdot b$ equals $a_1b_1 + a_2b_2 + \dots + a_nb_n$. This form of vector multiplication is the basic operation of the enciphering algorithm in the trapdoor knapsack system. To encipher the string of binary numbers that represents his message the sender first breaks the string into blocks of n bits, and for each block $x = (x_1, x_2, \dots, x_n)$ he forms the dot product $C = a \cdot x$ of that block with the public enciphering vector a , that is, $C = a_1x_1 + a_2x_2 + \dots + a_nx_n$.

The sum C is the information the sender transmits over the insecure channel, so that any eavesdropper is confronted with the task of recovering x from C and the numbers a_1, \dots, a_n . In what follows it will be convenient to refer to the elements of a vector x as the x_i 's (or to the elements of a vector a as the a_i 's), where the values of i are taken to be the integers from 1 to n . Since each x_i is equal to either 0 or 1, one can see that the problem of recovering x from C is equivalent to solving the knapsack or subset sum problem for these values of



IN A PUBLIC-KEY CRYPTOSYSTEM there is no need of a secure channel for the distribution of keys. As is shown here, each receiver generates two distinct keys: a public key E for implementing the public enciphering procedure G and a secret key D for implementing the public deciphering procedure H , which is the inverse of G . The keys E and D are related in the sense that they serve to specify inverse transformations G_E and H_D , but given E it is computationally infeasible to derive D : computing D from E would require thousands or even billions of years on the largest computer imaginable. Hence the receiver may communicate his enciphering key E over an insecure channel, as is shown here, or even list it in a public directory without compromising his deciphering transformation. A person who wants to send the plaintext P to the receiver operates on it with the receiver's enciphering transformation G_E to generate a ciphertext C , or $G_E(P)$. This ciphertext is transmitted over an insecure channel, and the receiver operates on it with the deciphering transformation H_D to recover the plaintext P , or $H_D(C)$. As long as the deciphering key D is kept secret there is no way for an eavesdropper to decipher the transmitted message. The challenge in designing such a system is to find general procedures G and H for which pairs of inverse keys E and D are easily generated but for which it is computationally infeasible to compute D from E . A source of such pairs is a group of mathematical problems that are said to be in the class NP (see illustration on opposite page).



continued on page 21

NEW APPLE][SOFTWARE

All Programs written in Integer BASIC for 16K

Cassette Disk

-SOFTBALL/TANK

\$14.95 \$18.95

by Dave Redhed

Two Arcade Games in Low Res for two players.

-CHARACTER TRAITS/PSALMS

\$14.95 \$18.95

by Dave Redhed

Two classic CAI programs. CT is based on the card game Character Clues.

Psalms are scriptural quotations dependent on your mood.

-TOUCH TYPING TUTOR

\$14.95 \$18.95

by Bill Massena

Improve your typing. Four lessons. Three preprogrammed lessons; Finger builders, General Typing, and BASIC Language. One you can input yourself. Tracks your speed and error count

-FOLLOW THAT TUNE

\$11.95 \$15.95

by Bill Massena

A musical game of "Simon Said." For up to 4 players. See your Dealer or Write:



COMPUTER SERVICES COMPANY
14109 S.E. 168th St.
RENTON, WA 98055

Washington residents add 5.3% sales tax



STOP SOFTWARE THIEVERY

Are you planning to market software on Apple® Diskettes?

Our customized anti-ripoff methods can protect your programs against bootleg copies and software piracy.

METHOD 1 protects software selling for up to \$50. Your programs cannot be LOADED then SAVED, or copied by any currently available COPY program.

METHOD 2 a DELUXE 5 point protection plan for software selling from \$50 to \$500.

METHOD 3 combines METHOD 2 software protection with sophisticated hardware protection for software selling for more than \$500.

We customize all methods to fit your needs. Write now for a free estimate. All correspondence is held strictly CONFIDENTIAL. Is your software already protected? Let us evaluate the cost effectiveness of your method. Non-disclosure agreements cheerfully accepted.

MICROMOTION

Microcomputers, Inc. - Micro
12077 Wilshire Blvd. Suite 506
West Los Angeles, CA 90025



When your Data Demands Security.

ENCODE/DECODE: #2 Software Security System for CP/M* & TRS-80# DOS

Let's face it, there is information which just isn't meant for everyone who uses or has access to your computer. Consider payroll or tax records. Until now, the only way to secure these and other valued or privileged records meant either "pulling the plug" or locking the discettes in a safe. Who wants to run to the safe each time an update needs to be made? At last a simple, effective and convenient method of data security is available—ENCODE/DECODE.

ENCODE/DECODE is a complete software security system for your micro/mini computer. ENCODE/DECODE can provide both the level of security and privacy you desire without loss of ON-LINE immediate access to data. ENCODE/DECODE is a sophisticated coding program which transforms data stored on disc into coded text which is completely unrecognizable. When it's time to access the file, it is decoded and ready for use. This means that data can be on-line and current with all your other files, yet only the user defined combination can retrieve it.

Multiple Security levels: Using ENCODE/DECODE you can easily maintain several layers of security through the use of separate combinations. This means that each file can have its own 'password' allowing only those with the 'password' access to the file.

ENCODE/DECODE uses a complex coding algorithm which supports over 987,000,000 possible combinations thus making accidental or 'exhaustive search' methods of decoding virtually impossible. Briefly, an encoded data file will appear scrambled and completely unintelligible until you decode it. Both encoding and decoding require the user defined combination.

Uses for ENCODE/DECODE are unlimited. Below are a few examples:

data bases	general ledger	inventory
payroll files	correspondence	accounts payable/
programs	tax records	receivable
text	mail lists	& more

ENCODE/DECODE is available in two versions. ENCODE/DECODE I provides a level of security suitable for normal use. ENCODE/DECODE II provides enhanced security for the most demanding needs. Both versions come supplied on discette and with a complete user's manual.

ENCODE/DECODE I: \$ 50.00

ENCODE/DECODE II: \$100.00

manual for above: \$ 15.00

Minimal system requirements: 24K CP/M; 16K disc for TRS-80

formats: CP/M 8" SOFT SECTORED, NORTHSTAR CP/M AND TRS-80 DOS

All Orders and General Information:
SUPERSOFT ASSOCIATES
P.O. BOX 1628
CHAMPAIGN, IL 61820
(217) 344-7596

Technical Hot Line: (217) 384-0847
(answered only when technician is available)

*CP/M REGISTERED TRADEMARK DIGITAL RESEARCH
#TRS-80 TRADEMARK TANDY CORP

OEM and dealer inquiries invited, overseas orders add \$5.00 shipping.

Circle 94 on inquiry card.

ould be a mistake
ception in this
onsider it a polit-
To have made an
uld have been a
ision," Brown

l Rev

Brown's no ex-
e, CDC pointed
proposal appar-
a threat to na-
several months
D approved the
ved the proposal
ional Security
DOD to take

ED

to 60 CPS. The
plugs into the
nt at its "catch-

its printing job.
ired. Full year

(312) 677-6080



decision, but the White House
would neither confirm nor
deny that. A White House
spokesman said President

in France.

DOD claims it will be differ-
ent this time. A CoCom deci-
sion to block sale of the CPU

can supply [k-
can't run a
don't know
he added.

Patent Granted for Microprocessor To Encrypt Software on a Chip

By Brad Schultz
CW Staff

SEATTLE — A means of
thwarting software piracy —
the illicit copying and black-
marketing of proprietary code
— may be available soon as a
microprocessor chip.

Robert M. Best of Seattle
was recently granted U.S. pat-
ent 4,168,396 for his design of
a "crypto-microprocessor"
(CM) that decrypts and exe-
cutes software.

Piracy can be a lucrative en-
terprise if the plundered soft-
ware runs on a large number
of installed CPUs without
needing significant modifica-
tion. The problem is acute for
vendors and for users that
routinely ship programs re-
presenting considerable in-
vestments.

Some user software, such as
computer-aided design and
manufacturing (CAD/CAM)
routines, offer major competi-
tive advantages — as long as
they are not intercepted by ri-
val firms.

Micro Modifications

Best's CM scheme entails
modifying existing microcom-
puters, but the changes would
not dampen their performance
significantly, he maintained.

Following the scheme, soft-
ware houses would encrypt
programs before delivering
them to customers. Each cus-
tomer would have a CM in-
stead of an ordinary micropro-
cessor in the systems sup-
ported by the program.

The CM would feature a cir-
cuit dedicated to decrypting
each instruction of the en-
crypted program as that in-
struction was fetched for exe-
cution, Best explained.

The CMs disseminated to the
program's official user base
and the device used by the
software house to encrypt the
program would all have equi-

valent encryption keys. This
would allow bona fide users to
exchange or remotely share
access to the program, Best
pointed out.

But prospective pirates could
not decipher the software un-
less they also obtained the ap-
propriate CM. By tapping
transmission lines or hijacking
shipments of the program, pi-
rates could only derive incom-
prehensible encrypted data.
The CMs could not output
comprehensible decrypted
listings of the routine.

Pirates and legitimate users
could make multiple copies of
the encrypted program deliv-
ered by the software house.
But only an authorized user
could run those copies, Best
emphasized.

Therefore, pirates could not
run copies of the program on
other microprocessors, nor
disassemble copies for use in
competing software products.
Proprietary data files could
also be safely distributed in ci-
pher along with a CM unit to
anonymous users, Best con-
tinued.

Best's patent describes sev-
eral encryption methods, in-
cluding a polyalphabetic sub-
stitution cypher that report-
edly employs a number of
small tables of randomly
generated bits stored on the
CM chip. "Each deciphering
cycle overlaps a bus-ad-
dressing cycle, so additional
clock cycles are not required,"
Best observed.

That means CMs may prove
competitive in performance
with conventional micropro-
cessors, he asserted.

Currently employed full-
time as a Cobol programmer,
Best is hunting in his spare
time for a semiconductor man-
ufacturer to implement the
CM design. "Production pro-
cessors could be available
within two years by modify-

ing existing microprocessor
designs," he indicated.

Best reportedly has 14 basic
patent applications pending in
the U.S., Japan, the UK and
other nations for a second
type of CM that would exe-
cute programs encrypted in
8-byte blocks in compliance
with the federal Data Encryp-
tion Standard.

SHOCKWATCH®



For installations
where system
security and data
integrity are
more than mere
buzz words.

Shockwatch® is a precise impact detector which can
prevent the use of damaged disk packs and cartridges
Media Recovery, Inc., 1435 Roundtable, Dallas, TX 75247
Call toll free: 1-800-527-9497.



OMSI PASCAL™

PDP-11 and LSI-11
Fast • Efficient • Supported

A full standard compiler for
RT-11, RSX-11M/D, and RSTS/E.
Features embedded assembler
code, FORTRAN interface, over-
lays, direct memory access, and
interactive symbolic debugger.
In use since 1975 — 300 users.

Oregon
minicomputer
Software inc.

2340 SW Canyon Road Portland, Oregon 97201
(503) 226-7760 TWX 910-464-4779

POP, LSI-11, RT-11, RSX-11, and RSTS are trademarks of
Digital Equipment Corporation, Maynard, Massachusetts.

10/22/79

COMPUTER WORLD